Course: Introduction to Geometric Modeling (ECS 178), WQ 2008
Professor: Dr. Bernd Hamann

Project 3: Interpolation and B-Splines
Date due: Friday, Feb. 15, 2008; demos in the afternoon

The third project requires the implementation of algorithms for **C^1- and C^2-continuous interpolation** and **B-spline curves**. Your program(s) must be developed on a workstation (or PC or laptop) using the OpenGL (or a similar) library for drawing curves and surfaces—and for providing user interface techniques (buttons, sliders, etc.). You are allowed to use the OpenGL library, X windows, Motif, or the Forms library. A user menu must be provided to **interactively specify all the parameters** required for the various algorithms.

a.) Implement the 2D version of the **C^2-continuous piecewise cubic interpolation** scheme discussed in class (computing the "de Boor" points $\mathbf{d}_i$ first, then constructing the Bézier points for each cubic curve segment). A user specifies the $(n+1)$ 2D points $\mathbf{p}_i$, $i = 0, ..., n$, to be interpolated. He/she can choose between uniform, chord length, or centripetal parametrization. Your program computes the $(n+3)$ points $\mathbf{d}_i$, $i = -1, ..., (n+1)$, and then constructs the Bézier points $\mathbf{b}_j$, $j = 0, ..., 3n$. Your program must display the polygon defined by the points $\mathbf{d}_i$, the polygon defined by the Bézier points $\mathbf{b}_j$, and the resulting interpolating curve.

b.) Implement a 2D version of the **de Boor algorithm** for evaluating a B-spline curve. A user specifies the order k and the de Boor control points $\mathbf{d}_i$, $i = 0, ..., n$. Your program computes the initial knot sequence $u_0 \leq u_1 \leq ... \leq u_{n+k}$ based on the Euclidean distances between the de Boor points, as discussed in class. Your program displays the original control polygon, the entire B-spline curve defined over the interval $[u_{k-1}, u_{n+1}]$ (evaluated at a user-specified resolution), and the knot sequence in form of a "knot bar." A user must be able to interactively change order, control points, and knots of a B-spline curve.

c.) Write a program for performing **C^1-continuous quadratic B-spline curve interpolation**. A user specifies a sequence of 2D points to be interpolated, and your program computes the sequence of knots and de Boor control points. The computation of the control points should follow the principle discussed in class: Compute a first control polygon by prescribing the derivative vector at the first point to be interpolated. Compute a second control polygon by prescribing a derivative vector at the last point to be interpolated. Finally, average these two control polygons to obtain the final polygon. The final control polygon and the knot sequence must be displayed along with the interpolating quadratic B-spline curve.

A user must be able to change all parameters easily by providing window areas used for the display and manipulation of parameters. Regarding the manipulation of control points (or points to be interpolated), a user must be able to **pick such points**, **change their positions in a window**, and **insert/delete such points**. If two or more points are within the same "pick" region, your program must select the point closest to the center of the "pick" region. To move a point in a window use the left and right mouse buttons to change its x- and y-coordinates.

Besides having to demonstrate your program, prepare a short, about one-page "user's manual" explaining how to use your program.

DO NOT REMOVE YOUR PROGRAM!!! YOU CAN USE IT FOR FURTHER ASSIGNMENTS.

HAVE FUN!!!